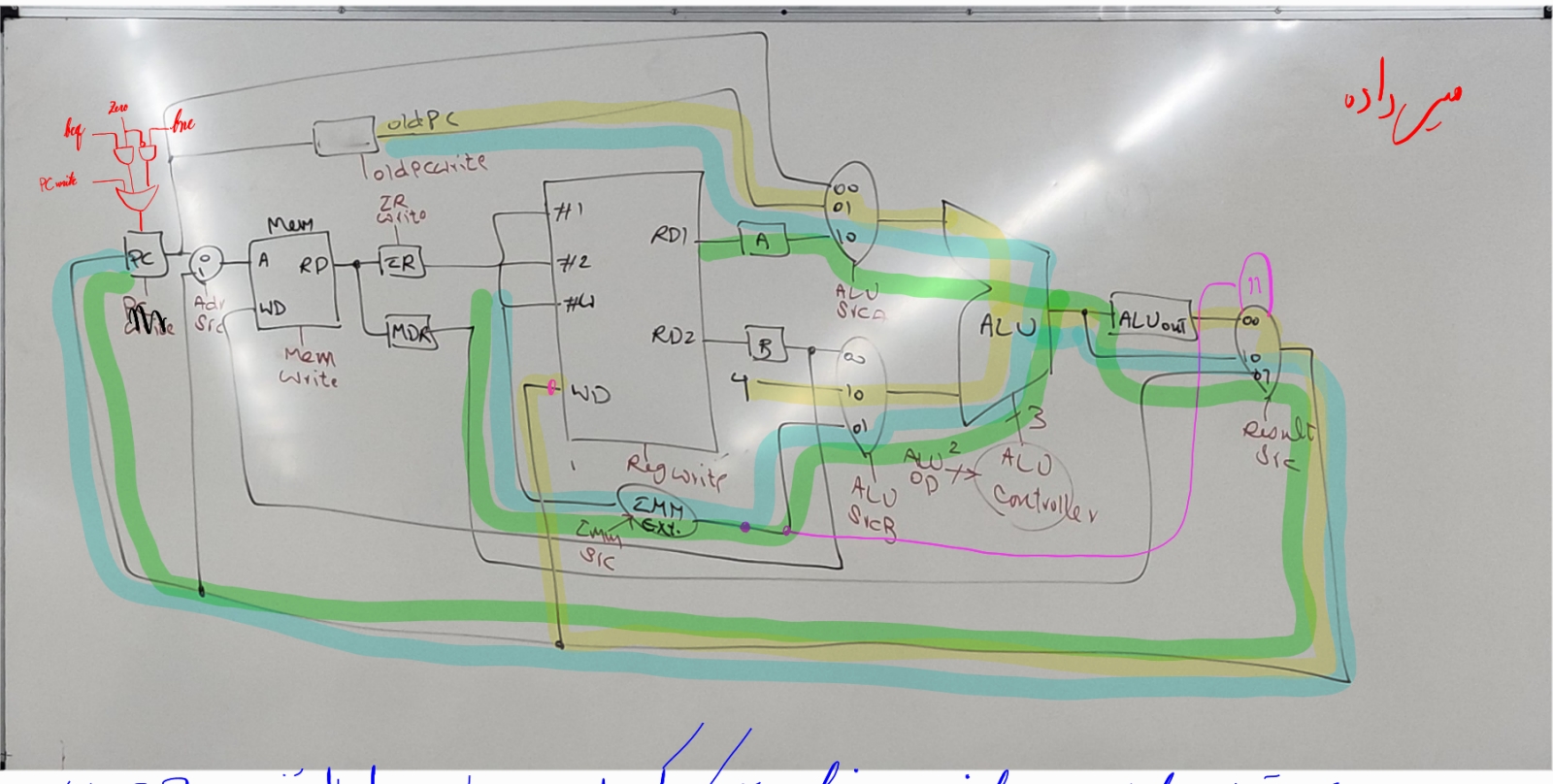
"بسم الله الرحمن الرحیم"

پروژه سوم معماری کامپیوتر

سید مانی پرستگانی ۸۱۰۸۰۱۰۸۵
معین الگری شهر ۸۱۰۸۰۱۰۵۵



میرداده

تنها دستورات jalr، jal و lui هستند که ممکن است نیاز به اعمال تغییر در DP داشته باشند.

بررسی jal:

Jal rd, آدرس بیتی 20 → { rd ← pc+4 ; pc ← pc + آدرس بیتی 20 }

DP نیاز به تغییر ندارد!

بررسی jalr:

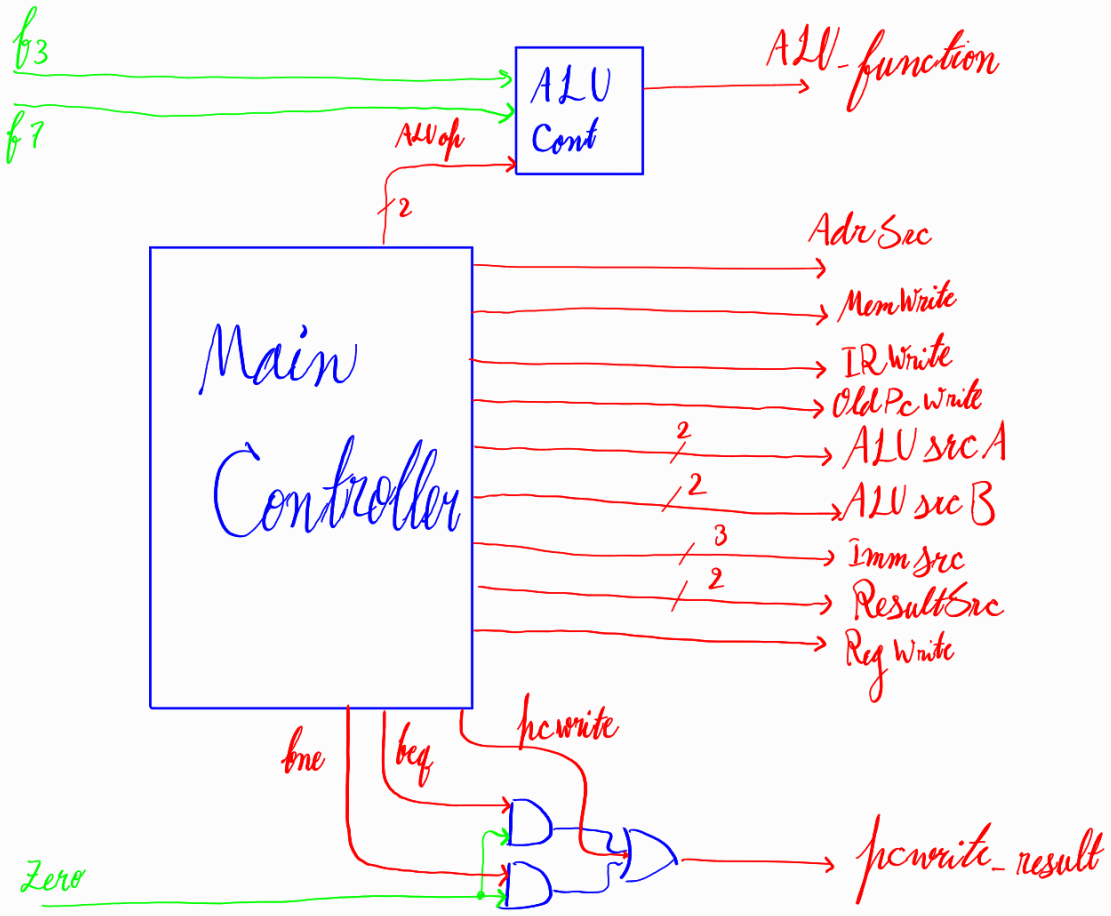
Jalr rd, rs, آدرس بیتی 12 → { rd ← Pc+4 ; PC ← rs + آدرس بیتی 12 }

بررسی lui:

lui rd, Imm20

تنها کافی است از Imsrc به ورودی WD از RF پس (MUX) بگذاریم.


طراحی / تغییرات واحد کنترل
lui
9
ImmSrc = 100
RegWrite = 1
ResultSrc = 11
I-Type
ALUSrcA = 10
ALUSrcB = 01
ImmSrc = 000 (I-Type)
ALUop = 11 (I-Type)
8
Jump
11
ALUSrcA = 01
ALUSrcB = 10
ALUop = 00
jal/jalr
12
RegWrite = 1
ResultSrc = 00
SaveReturn Adr
jalr
jal
Jal
ResultSrc = 10
PCwrite = 1
ALUSrcA = 01
ALUSrcB = 01
ImmSrc = 011
ALUop = 00
13
Jalr
ResultSrc = 10
ALUop = 00
ALUSrcA = 10
ALUSrcB = 01
PCwrite = 1
ImmSrc = 000 (I-Type)
14
rs = rd + 512
IF
AdrSrc = 0
IRwrite = 1
ALUSrcA = 00
ALUSrcB = 10
ALUop = 00
ResultSrc = 10
PCwrite = 1
old PCwrite = 1
0
ID
ALUSrcA = 01
ALUSrcB = 01
ALUop = 00 (+)
ImmSrc = 010 (beq)
1
mem
ref
MemRef
ALUSrcA = 10
ALUSrcB = 01
if (lw) ImmSrc = 000
if (sw) ImmSrc = 001
2
lw
MemRead
ResultSrc = 00
AdrSrc = 1
3
LoadWord
ResultSrc = 01
RegWrite = 1
4
sw
ResultSrc = 00
AdrSrc = 1
MemWrite = 1
5
RT
R-Type
ALUSrcA = 10
ALUSrcB = 00
ALUop = 10
6
B-Type
ALUSrcA = 10
ALUSrcB = 00
ALUop = 01 (-)
ResultSrc = 00
beq = (beq)
bne = (bne)
7
WriteBack
ResultSrc = 00
RegWrite = 1
10
beq
beq = 1



ALU cont:

ALUop	f3	f7	ALU func
00	X	X	000
01	X	X	001
10	add sub and or slt xor		000 001 010 011 100 101
11	addi ori ⋮	X	

Imm src:

Imm src		Type
000	{{20{I[31]}}, I[31:20]}	I-Type
001	{{20{I[31]}}, I[31:25], I[11:7]}	S-Type
010	{{19{I[31]}}, I[31], I[7], I[30:25], I[11:8], 1'b0}	B-Type
011	{{12{I[31]}}, I[19:12], I[20], I[30:21], 1'b0}	J-Type
100	{I[31:12], 12{1'b0}}	U-Type

Main Controller

opc	Inst	Reg write	Imm src	ALU src	mem write	Result src	branch	ALU op	Jump	jalr
51	R-T	1	XXX	0	0	01	0	10	0	0
3	lw	1	000	1	0	01	0	00	0	0
19	I-T	1	000	1	0	00	0	11	0	0
sw 35	S-T	0	001	1	1	XX	0	00	0	0
jal 111	J-T	1	011	X	0	10	0	XX	1	0
99	B-T	0	010	0	0	XX	1	01	0	0
lui 55	U-T	1	100	X	0	11	0	XX	0	0
103	jalr	1	000	1	0	00	0	00	0	1

*Assembly Code:

```
int m = A[0];
for (int i=0; i<20; i++)
    if (m < A[i])
        m = A[i];
```

(s1 → m, s0 → A[0], s2 → i)

```
1  main:
2      addi s0, zero, 0 # 0 is the start of array
3      lw s1, 0(s0)
4      addi s2, zero, 0
5      addi s0, s0, 4
6  loop:
7      slti t0, s2, 10
8      beq t0, zero, end_loop
9      lw t0, 0(s0)
10     slt t1, s1, t0
11     bne t1, zero, else
12     add s1, t0, zero
13 else:
14 addi s2, s2, 1
15 addi s0, s0, 4
16 jal loop
17 end_loop:
18 addi s4, zero, 0
```

jalr rd, rs, آدرس جامپ بیتی →
- rd ← Pc+4
- PC ← rs + آدرس جامپ بیتی

پاسخ نهایی s1 خواهد بود